

# Future Internet Enterprise Systems: A Flexible Architectural Approach for Innovation

Daniela Angelucci, Michele Missikoff, and Francesco Taglino

Istituto di Analisi dei Sistemi ed Informatica “A. Ruberti”

Viale Manzoni 30, I-00185 Rome, Italy

daniela.angelucci@gmail.com, michele.missikoff@iasi.cnr.it,  
francesco.taglino@iasi.cnr.it

**Abstract.** In recent years, the evolution of infrastructures and technologies carried out by emerging paradigms, such as Cloud Computing, Future Internet and SaaS (Software-as-a-Service), is leading the area of enterprise systems to a progressive, significant transformation process. This evolution is characterized by two aspects: a progressive commoditization of the traditional ES functions, with the ‘usual’ management and planning of resources, while the challenge is shifted toward the support to enterprise innovation. This process will be accelerated by the advent of FInES (Future Internet Enterprise System) research initiatives, where different scientific disciplines converge, together with empirical practices, engineering techniques and technological solutions. All together they aim at revisiting the development methods and architectures of the Future Enterprise Systems, according to the different articulations that Future Internet Systems (FIS) are assuming, to achieve the Future Internet Enterprise Systems (FInES). In particular, this paper foresees a progressive implementation of a rich, complex, articulated digital world that reflects the real business world, where computational elements, referred to as FInER (Future Internet Enterprise Resources), will directly act and evolve according to what exists in the real world.

**Keywords:** Future Internet, Future Enterprise Systems, component-based software engineering, COTS, SOA, MAS, smart objects, FInES, FInER.

## 1 Introduction

In recent years, software development methods and technologies have markedly evolved, with the advent of SOA [15], MDA [16], Ontologies and Semantic Web, to name a few. But there are still a number of open issues that require further research and yet new solutions. One crucial issue is the excessive time and cost required to develop enterprise systems (ES), even if one adopts a customisable pre-built application platform, e.g., an ERP solution.

This paper explores some emerging ideas concerning a new generation of Internet-based enterprise systems, along the line of what has been indicated in the FInES

(Future Internet Enterprise Systems) Research Roadmap<sup>1</sup>, a study carried out in the context of the European Commission, and in particular the FinES Cluster of the D4 Unit: Internet of Things and Enterprise Environments (DG InfSo). The report claims that we are close to a significant transformation in the enterprise systems, where (i) the way they are developed, and (ii) their architectures, will undergo a progressive paradigm shift. Such paradigm shift is primarily motivated by the need to repositioning the role of enterprise systems that, since their inception, have been conceived to support the management and planning of enterprise resources. Payroll, inventory management, and accounting have been the first application areas. Then, ES progressively expanded their functions and aims, but the underlying philosophy remained the same: supporting the value production in the day by day business, optimising operations and the use of resources, with some look ahead capabilities (i.e., planning). In the recent period there has been a clear movement towards a progressive commoditization of such traditional ES functions. This movement is further facilitated by the evolution of infrastructures and technologies, starting from Cloud Computing and Future Internet, and, on top of those, the Software-as-a-Service (SaaS) paradigm that is progressively providing new ways of conceiving and realising enterprises software applications.

In essence, while enterprise management and planning services will be increasingly available from the ‘cloud’, in a commoditised form, the future business needs (and challenges) are progressively shifting towards the support to enterprise innovation. But also innovation cannot remain as it used to be: Future Internet, Web 2.0, Semantic Web, Cloud Computing, SaaS, Social Media, and similar emerging forms of distributed, open computing will push forward new forms of innovation such as, and in particular, Open Innovation [3]. The quest for continuous, systematic business innovation requires (i) ES capable of shifting the focus to ideas generation and innovation support, and (ii) new agile architectures, capable of (instantly) adjusting to the continuous change required to enterprises. New business requirements that current software engineering practices do not seem to meet. Therefore we need to orientate the research towards new ES architectures and development paradigms, when the role of ICT experts will be substantially reduced. To this end, we wish to propose three grand research challenges.

The first grand research challenge (GRC) implies for ICT people to surrender the mastership of ES development, handing it over to business experts. To this end, the ICT domain needs to push forward the implementation of future ES **development environments**, specifically conceived to be directly used by business experts. Such development environments will be based on an evolution of MDA, being able to separate the specification and development of the (i) strategic business logic from the (ii) specific business operations and, finally, their (iii) actual implementation. A central role will be played by enterprise system Business Process Engineering, for the above point (i), and a new vision, based on a new family of reusable components, in the implementation of enterprise operations (and related services) automation, for the last two points. Reusable components mash-up techniques, and advanced graphical user

---

<sup>1</sup> <http://cordis.europa.eu/fp7/ict/enet/documents/task-forces/research-roadmap/>

interfaces will foster new development environments conceived for business experts to directly intervene in the development process.

The second grand research challenge concerns the **architecture** of the Future Internet Enterprise Systems (FInES) that need to deeply change with respect to what we have today. A new paradigm is somehow already emerging nowadays, pushed by the new solutions offered in the Future Internet Systems (FIS) field. In particular, we may mention: the Internet of Services (IoS), Internet of Things (IoT) and *smart objects*, Internet of Knowledge (IoK), Internet of People (IoP). But these solutions need to further evolve towards a better characterisation in the business direction, allowing different aspects of the business reality (functions, objects, actors, etc.) to acquire their networked identity, together with a clear and precise definition (i.e., science based) of their (information) structure, capabilities, and mutual relationships. In fact, what is missing today is a unifying vision of the disparate business aspects and entities of an enterprise, supported by an adequate theory, able to propose new technological paradigms. The idea is that all possible entities within (and outside) an enterprise will have a digital image (a sort of ‘avatar’) that has been referred to as Future Internet Enterprise Resource (FInER) in the FInES Research Roadmap. So, the second grand research challenge consists in conceiving new, highly modular, flexible FInERs for the FInES architectures to be based on.

Finally, there is a third grand research challenge, that of shifting the focus of the attention from the management and planning of business and enterprise resources to **enterprise innovation**. This GRC requires, again, a strategic synergy between ICT and business experts. Together, they need to cooperate in developing a new breed of services, tools, software packages, interfaces and user interaction solutions that are not available at the present time. A new family of ICT solutions aimed at supporting the conception, design, implementation and deployment of enterprise innovation, including assessment of impact and risks.

In this paper, we intend to further elaborate on these challenges. In particular on the first and the second GRC that concern the development of new FInESs capable of offering to the business experts the possibility of directly governing the development of software architectures. This will be possible if such software architectures will correspond to the enterprise architectures, and will be composed by elements tightly coupled with business entities. The achievement of this objective relies on a number of ICT solutions that are already emerging: from Cloud Computing to Social Media, to Service-oriented Computing, from Business Process Engineering to semantic technologies and mash-up. An exhaustive analysis of the mentioned technologies is outside the scope of this paper, below we will briefly survey some of them.

The rest of the paper is organised as follows. Next Section II provides an overview of the evolution of the main technologies that, in our vision, will support the advent of the FInES. Section III presents the main characteristics of a FInES innovation-oriented architecture. Section IV introduces the new component-oriented approach based on the notion of a FInER, seen as the new frontier to software components aimed at achieving agile system architectures. Section V provides some conclusions and a few lines that will guide our future work.

## 2 A Long March towards Component-Based Enterprise Systems

FIInES represents a new generation of enterprise systems aimed at supporting continuous, open innovation. Innovation implies continuous, often deep changes in the enterprise; such changes must be mirrored by the enterprise systems: if the latter are too complex, rigid, difficult to evolve, they will represent a hindering factor for innovation. A key approach to system flexibility and evolvability is represented by highly componentized architectures.

Traditionally, the software engineering community has devoted great attention to design approaches, methods and tools, supporting the idea that large software systems can be created starting from independent, reusable collections of pre-existing software components.

This technical area is often referred to as Component Based Software Engineering (CBSE). The basic idea of software componentization is quite the same as software modularization, but mainly focused on reuse. CBSE distinguishes the process of "component development" from that of "system development with components" [9].

CBSE laid the groundwork for the Object Oriented Programming (OOP) paradigm that in a short time imposed itself over the pre-existing modular software development techniques. OOP aims at developing applications and software systems that provide a high level of data abstraction and modularity (using technologies such as COM, .NET, EJB and J2EE).

Another approach to componentization is that of the Multi Agent Systems (MAS), which is based on the development of *autonomous, heterogeneous, interacting* software agents. Agents mark a fundamental difference from conventional software modules in that they are inherently autonomous and endowed with advanced communication capability [10].

On the other side, the spread of the Internet technologies and the rising of new communication paradigms, has encouraged the development of loosely coupled and highly interoperable software architectures through the spread of the Service-Oriented approach, and the consequent proliferation of Service-Oriented Architectures (SOA). SOA is an architectural approach whose goal is to achieve loose coupling among interacting software services, i.e., units of work performed by software applications, typically communicating over the Internet [11].

In general, a SOA will be implemented starting from a collection of components (e-services) of two different sorts. Some services will have a 'technical' nature, conceived to the specific needs of ICT people; some other will have a 'business' nature, reflecting the needs of the enterprise. Furthermore, the very same notion of an e-service is an abstraction that often hides the entity (or agent) that in the real world provides such a service. Such an issue may seem trivial to ICT people (they need a given computation to take place; where it is performed or who is taking care of it is inconsequential). Conversely, for business people, services are not generated 'in the air': there is an active entity (a person, an organization, a computer, a robot, etc.) that provides the services, with a given cost and time (not to mention SLA, etc.), associated to it.

In summary, Web services were essentially introduced as a computation resource, transforming a given input to produce the desired output, originally without the need to have a persistent memory and an evident state. Such a notion of ICT service is very specific and, as such, not always suited when we consider business services, where states, memories, and even the pre-existing history of the entity providing the business service, are important.

Our aim to achieve an agile system architecture made up of FInERs put its basis upon the spread of the Cloud Computing philosophy, but revising and applying it into the specific context of developing new FInESs, where business expert can directly manage a new generation enterprise software architectures. Cloud Computing represents an innovative way to architect and remotely manage computing resources: this approach aims at delivering scalable IT resources over the Internet, as opposed to hosting and operating those resources (i.e. applications, services and the infrastructure on which they operate) locally. It refers to both the applications delivered as services over the Internet and the hardware and system software in the datacenters that provide those services [12]. Cloud Computing may be considered the basic support for a brand new business reality where FInERs can easily be searched, composed and executed by a business expert. FInERs will implement a cloud-oriented way of designing, organizing and implementing the enterprises of the future.

In conclusion, for decades component technologies have been developed with an ICT approach, to ease software development processes. Conversely, we propose to base a FInES architecture on building blocks based on business components. In addressing a component-based architecture for FInES, we felt the need for a new component-oriented philosophy that should be closer to, and make good sense for, business people. The idea of a FinER goes in this direction. From an ICT perspective, the notion of a FInER integrates many characteristics of the OOP, MAS, and SOA just illustrated, plus the key notion of *smart object*. But the key issue is that a FInER can exist only if it represents a business entity and is recognized as such by business people.

### 3 Guidelines for a FInES Architecture

With the idea of a continuous innovation process going on in parallel to the everyday business activities, a FInES needs to integrate the two levels: doing business and pushing forward innovation, constantly evolving along a loop similar to that represented in Fig.1. From the architectural point of view, a FInES is seen as a federation of systems relaying on two major infrastructures: one for the advanced management of knowledge and the other for semantic interoperability. The figure reports four systems identified by the same numerals reported in the FInES Research Roadmap, but the prefix here is the letter 'S' to indicate the *systems* and 'I' to indicate the *infrastructures* (instead of 'RC' for research challenge used in the FInES Research Roadmap).

The macro-architecture is briefly described below along the lines of what is reported in the cited research roadmap (where additional details can be found).

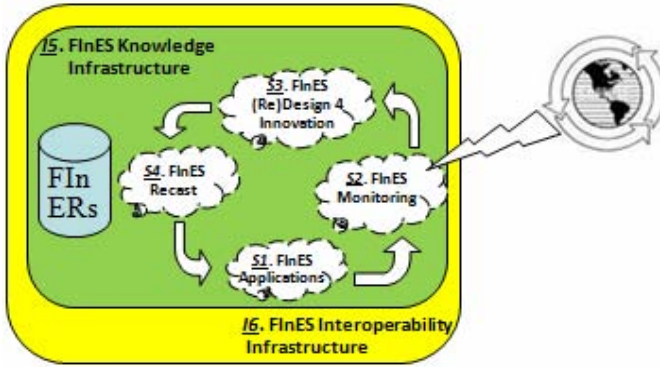


Fig. 1. FInES Macro-architecture

### S1 – FInES Open Application System

This system is devoted to the business operations for the day by day business and value production, with planning capabilities. In terms of functionalities it is similar to an ERP as we know it today, but its architecture is inherently different, since it is built by business experts by using Enterprise Systems/Architectures (including Business Process) Engineering methods and tools starting from a repository of FInERs, the new sort of computational enterprise components just introduced (see below for more details).

### S2 – FInES Open Monitoring System

This system is dedicated to the constant monitoring and assessment of the activities of S1, to keep under control the health of the enterprise, its performances, both internally (HR, resources, productivity, targets, etc.) and with respect to the external world (markets, competitors, natural resources, new technologies, etc.). S2 is able to signal when and where an innovation intervention is necessary, or even suitable.

### S3 – FInES Re-design System

This system is mainly used by business experts who, once identified the area(s) where it is necessary / suitable to intervene, proceed in designing the interventions on the enterprise and, correspondingly, on FInES. This task is achieved by using a platform with a rich set of tools necessary to support the business experts in their redesign activities that are, and will remain, largely a ‘brain intensive’ job.

### S4 – FInES Recast System

This system has the critical task of implementing the new specifications identified and released by S3. The activities of S4 can be seen roughly divided in two phases. The first phase is focused on the identification and acquisition of the new components to be used (e.g., the new FInERs). The second phase is the actual deployment of the new FInES. Updating a FInES is a particularly delicate job, since the changes need to be achieved without stopping the business activities.

Finally, the proposed architecture relies on two key infrastructures: the FInES **Knowledge Infrastructure** (I5-FKI) aimed at the management of the information and knowledge distributed within the different FInERs, maintaining a complex, net-



worked structure, conceived as an evolution of the Linked Open Data<sup>2</sup> of today; and the FInES **Interoperability Infrastructure** (I6-FII), supporting the smooth communication among the great variety of components, services, tools, platforms, resources, (produced by different providers) that compose a FInES.

## 4 The New Frontier for ES Components: The FInER Approach

In FInES architectures we intend to push the component-oriented engineering to an extreme, applied such an approach both horizontally (i.e., for different classes of applications and services) and vertically (using sub-parts at different levels of granularity). But the most relevant aspect is the large role played by a new sort of components: FInERs, i.e., computational units representing enterprise entities. They are recognised by business people as constituent parts of the enterprise, and therefore easily manipulated by them. A FInER has also a computational nature, characterised by 5 aspects, as described below.

A FInER can be a concrete, tangible entity, such as a drilling machine or an automobile, or intangible, such as a training course, a business process, or a marketing strategy. A complex organization, such as an enterprise, is itself a FInER. FInERs are conceived to interact and cooperate among themselves, in a more or less tight way, depending on cases (alike to real world business entities do).

To be more precise, we define a FInER as a 5-tuple:

$$F = (FID, GR, M, B, N)$$

The defining elements can greatly vary, depending on the complexity of the enterprise entity represented. In general we have:

*FID: FInER identifier.* This is a unique identifier defined according to a precise, universally accepted standard (e.g., according to IPv6, URI<sup>3</sup>, or ENS<sup>4</sup>).

*GR: Graphical Representation.* This can vary from a simple GIF to a 3D model, to a JPEG representation, including possibly a video.

*M: Memory.* The FInER memory again can be simply a ROM with basic info (e.g., its sort, date of production, etc.) to complex knowledge about its components, properties and the history of its lifecycle.

*B: Behaviour.* The description of the functional capability of the FInER. It structured on the line of the profile (IOPE: Input, Output, Processing, Effects) and model (essentially, a set of workflows) of OWL-S [13]. Besides, it may have self-monitoring capabilities.

*N: Networking.* The specification of all the possible interactions the FInER can achieve, with the protocols for issuing (as client) or responding (as server) to request messages. It is structured according to the grounding of OWL-S.

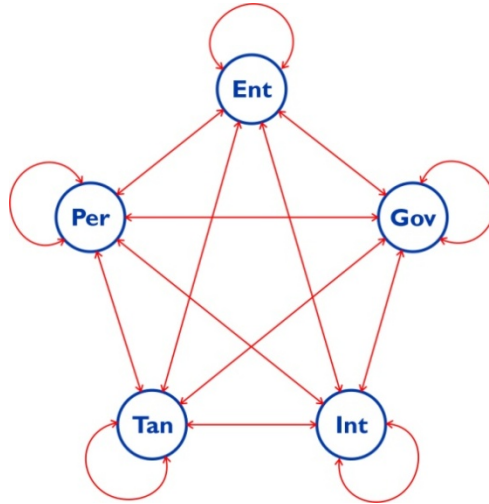
<sup>2</sup> <http://esw.w3.org/SweoIG/TaskForces/CommunityProjects/LinkingOpenData>

<sup>3</sup> Universal Resource Identifier, see Berners-Lee et al., 2005: <http://gbiv.com/protocols/uri/rfc/rfc3986.html>

<sup>4</sup> ENS: Entity Name System, proposed by the OKKAM project ([www.okkam.org](http://www.okkam.org))

As a first level, 5 FInERs categories have been identified:

- **Enterprise**, being the ‘key assembly’ in our work.
- **Public Administration**, seen in its interactions with the enterprise.
- **People**, a special class of FInERs for which avatars are mandatory.
- **Tangible entity**, from computers to aircrafts, to buildings and furniture.
- **Intangible entity**, for which a digital image is mandatory.



**Fig. 2.** The FInER Pentagone

All these FInERs will freely interact and cooperate, according to what happens for their real world counterparts. A complete interaction scheme is sketchily represented by the FInERs Pentagone in Fig. 2.

The proposed approach foresees a progressive implementation of a rich, complex, articulated digital world where computational elements, i.e. FInERs, will largely reflect what exists in the real (analogical) world. As a consequence, we expect that the FInER world will include all possible objects, creatures, entities, both simple and complex, animated and inanimate, tangible and intangible, that can be found in the real world.

As previously said, the FInER entities will have a unique identity and will be constantly connected (transparently, in a wired or wireless mode) to the Internet, to reach other FInERs, but at the same time to be themselves reachable anytime, anywhere, by any other FInER.

## 5 The FInES Approach to Design and Runtime Operations

As anticipated, the key platforms in the adoption of the FInES approach are the (re)design and development, and the runtime execution platforms.



### 5.1 A Business-Driven FInES Development Platform

In order to put the business experts at the centre of the ES development process, we foresee a platform where FInERs are visualised and directly manipulated by the user, while they reside in the Cloud and are reached through the Internet. On the FInES development environment (see Fig. 3), FInERs are visually represented in a 3D space that models the enterprise reality (i.e., a Virtual Enterprise Reality) where the user can navigate and manage changes. At a lower level, simpler FInERs will be aggregated to form more complex ones. The composition will take place in a partial automatic and bottom-up way, since many FInERs have autonomic capabilities and know already how to aggregate and connect to each other. Then, business experts supervise and complete the work. This approach represents a marked discontinuity with the past, since a FInES will be directly engineered by business experts and not by IT specialists. In fact, business experts will be able to select, manipulate, and compose FInERs at best, since they know better than IT specialists what the different business entities (represented by FInERs) are, which characteristics they have, how they can be connected one another to cooperate for achieving successful business undertakings. Future Internet will play a central role in supporting the discovery of the needed FInERs that often will be virtually acquired (in case of intangible assets), since they will be positioned in different parts of the enterprise or in the Cloud, depending on the cases.

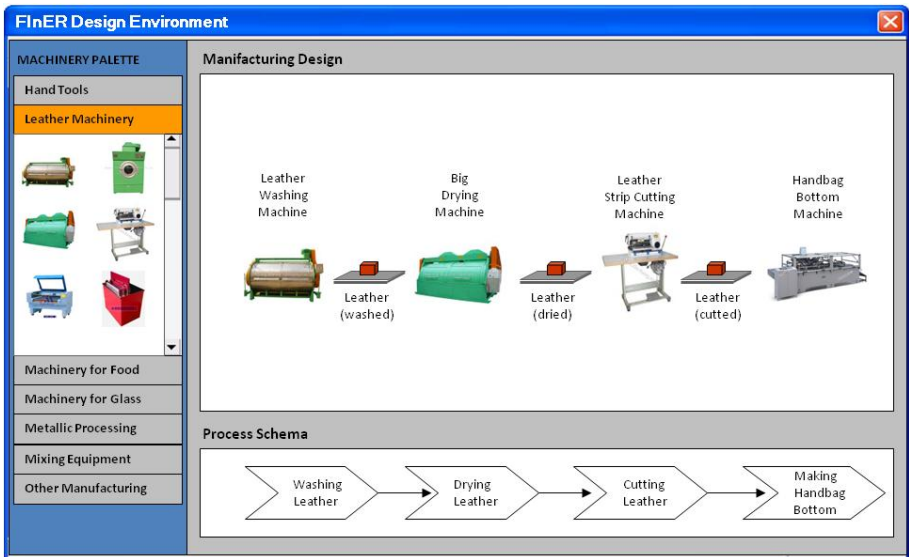


Fig. 3. FInES design environment

### 5.2 A Cloud-Based Architecture for FInERs Runtime

Once a FInES has been assembled (or re-casted, see Fig. 1), a runtime environment will recognise, connect, and support the execution and collaboration of the FInER components. Since FInERs will be developed by different suppliers, using different

approaches and techniques, one of the key aspects here is represented by the availability of a number of Smart Interoperability Enablers capable of recognising and transmitting events and data generated during FInERs' operations. There is not a centralised database, the information will stay by the business entity to which they pertain or in the Cloud. A similar interface, representing a Virtual Enterprise Reality, will be made available to the users during business operations to navigate in the enterprise and see how the operations evolve.

The computational resources of a FInES are maintained in the Computing Cloud, and are recursively linked to compose complex FInERs starting from simpler ones. Fig. 4 reports a three levels macro-architecture where the top level is represented by the real world, with the enterprise and the actual business resources. Below we distinguish *High Level* and *Low Level* and FInERs. At the bottom level, there are atomic and simple FInER aggregations that do not represent yet consistent business entities. E.g., a simple production machine, with its components, elementary operations belonging to one or more business process, a teaching textbook. The High Level FInERs are representatives of business entities that have a clear business identity and are aimed at the reaching a business goal. For instance, a production chain, a training course, a company department. (A formal definition of Low and High Level FInERs is difficult to be achieved, much depends on the business perspective, the industrial sector, etc.).

The platform is mainly activated by business events that are generated in the enterprise (but also the external world, in general) and trigger the execution of the FInERs that correspond to the business entities that need to respond to the triggering events. It is also possible to have pre-planned events that are generated by a FInER (e.g., the achievement of a specific operation or the reaching of a deadline) and propagate over the FInES according to a pre-defined semantics (e.g., a specific business process). In principle, all the FInERs are hosted in the Cloud.

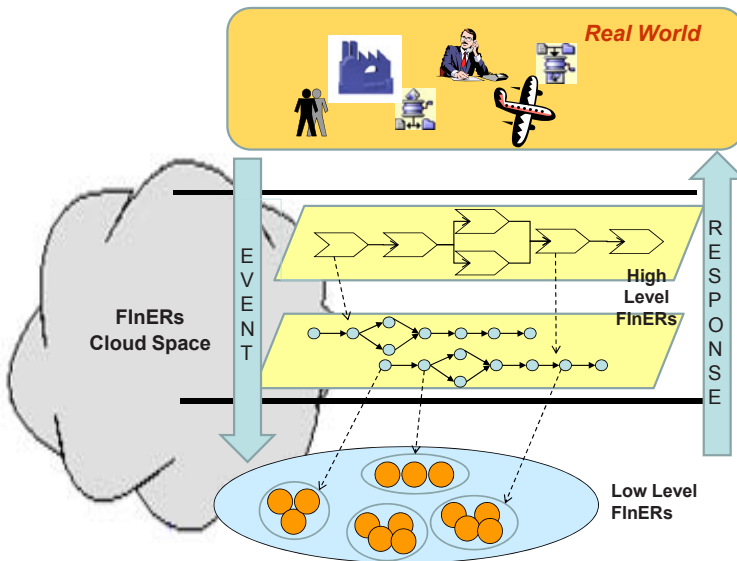


Fig. 4. FInES Runtime Environment

The runtime architecture of Fig. 4 is described in a sketchy way, aiming to highlight the main issues represented by (i) the highly modular structure, (ii) the mirroring of the real world business entities, (iii) the full control of the architecture by the business experts.

## 6 Conclusions

At the beginning of the 80s, the SUN had a visionary catchphrase, summarised in the sentence ‘The Network is the Computer’. As it happens with early intuitions, it took too long to happen and now only few remember this foretelling. Actually, current ICT achievements show that this is going to be fully achieved in a short term to go. As a next prophecy we propose “the Enterprise is the Computer”, meaning that an enterprise, with all its FInERs deployed and operational, will enjoy a fully distributed computing power, where computation will be directly performed by enterprise components, mainly positioned in the enterprise itself or in the Cloud (typically, in case of intangible entities). This approach represents a disruptive change, from both a technological point of view and a business perspective. In particular, in the latter, business people will be involved in building and maintaining large scale computing solutions simply interacting with a familiar (though technologically enhanced) business reality. Along this line, the (in)famous business / IT alignment problem will simply disappear, since the gap has been solved at its roots.

We are aware that the vision presented in this paper is a long term one, however, we believe that it will advance progressively and in the next years we will see an increasing integration of technologies that today are still loosely connected (e.g., IoT, IoS, Multi-Agent Systems, Cloud Computing, Autonomic Systems) and, in parallel, some key areas of the enterprise that will start to benefit of the FInES approach.

**Acknowledgment.** We acknowledge the FInES Cluster (European Commission), and in particular the FInES Research Roadmap Task Force for their contribution in producing much of the material reported in this paper.

**Open Access.** This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

## References

1. Bouckaert, S., De Poorter, E., Latré, B., et al.: Strategies and Challenges for Interconnecting Wireless Mesh and Wireless Sensor Networks. *Wireless Personal Communications* 53(3) (2010)
2. Buxmann, P., Hess, T., Ruggaber, R.: - Internet of Services. *Business & Information Systems Engineering* 1(5), 341–342 (2009)
3. Chesbrough, H.: *Open Innovation: The new Imperative for Creating and Profiting from Technology*. Harvard Business School Press (2003)

4. Luftmann, J.N., Papp, R., Brier, T.: Enablers and Inhibitors of Business-IT-Alignment. *Communications of AIS* 1(11) (1999)
5. Mansell, R.E.: Introduction to Volume II: Knowledge, Economics and Organization. In: Mansell (ed.), *The Information Society, Critical Concepts in Sociology*, Routledge (2009)
6. Cordis.lu: Proposition, Informal Study Group on Value. Retrieved May 29, 2010, from Value Proposition for Enterprise Interoperability Report (2009), [http://cordis.europa.eu/fp7/ict/enet/ei-isg\\_en.html](http://cordis.europa.eu/fp7/ict/enet/ei-isg_en.html)
7. Sykes, D., Heaven, W., Magee, J., Kramer, J.: From goals to components: a combined approach to self-management. In: *Proceedings of the 2008 international workshop on Software engineering for adaptive and self-managing systems* (2008)
8. Villa, F., Athanasiadis, I.A., Rizzoli, A.E.: Modelling with knowledge: A review of emerging semantic approaches to environmental modeling. *Environmental Modelling & Software* 24(5) (2009)
9. Crnkovic, I., Larsson, S., Chaudron, M.: Component-based Development Process and Component Lifecycle. In: *27th International Conference on Information Technology Interfaces (ITI)*, Cavtat, Croatia, IEEE, Los Alamitos (2005)
10. Nierstrasz, O., Gibbs, S., Tschritzis, D.: Component-oriented software development, Special issue on analysis and modeling in software development, pp. 160–165 (1992)
11. Petritsch, H.: Service-Oriented Architecture (SOA) vs. Component Based Architecture, white paper, TU Wien (2006), <http://whitepapers.techrepublic.com.com>
12. Armbrust, M., et al.: Above the Clouds: A Berkley View of Cloud Computing, EECS-2009-28 (2009)
13. Martin, D., et al.: Bringing Semantics to Web Services with OWL-S. In: *Proc. Of WWW Conference* (2007)
14. Clark, D., et al.: Newarch project: Future-generation internet architecture. Tech Rep. MIT Laboratory for Computer Science (2003), <http://www.isi.edu/newarch/>
15. Tselentis, G., et al. (eds.): *Towards the Future Internet- Emerging Trends from European Research*. IOS Press, Amsterdam (2010)
16. Papazoglou, M.P.: *Web Services: Principles and Technology*. Prentice-Hall, Englewood Cliffs (2007)
17. Mellor, S.J., Scott, K., Uhl, A., Weise, D.: Model-driven architecture. In: Bruel, J.-M., Bellahsene, Z., et al. (eds.) *OOIS 2002. LNCS, vol. 2426*, p. 290. Springer, Heidelberg (2002)